

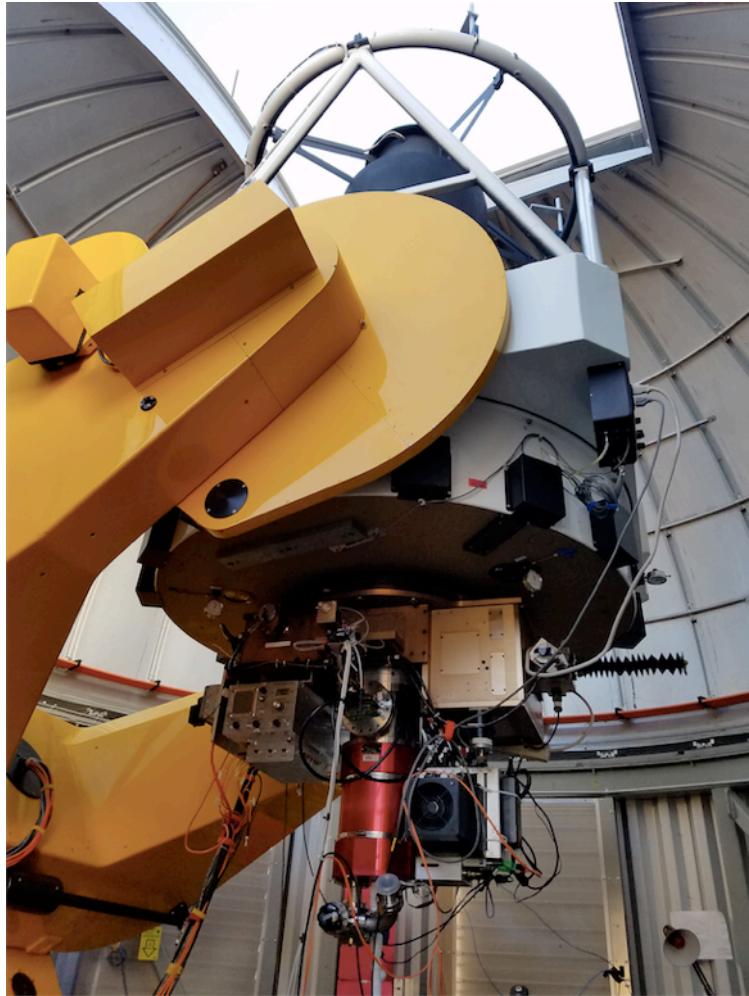
Monocam Fringing Update

Jason Brooks

Andrei Nomerotski

Merlin Fisher-Levine

Monocam at 1.3m



1.3m Telescope

- scale 40.0 arcsec/mm
- f/4.0
- FOV 0.44 x 0.44 sq.deg.
- Blank field CABlank5
 - RA 16 24 33
 - DEC +55 43 59
- y4 filter

Image Specifications

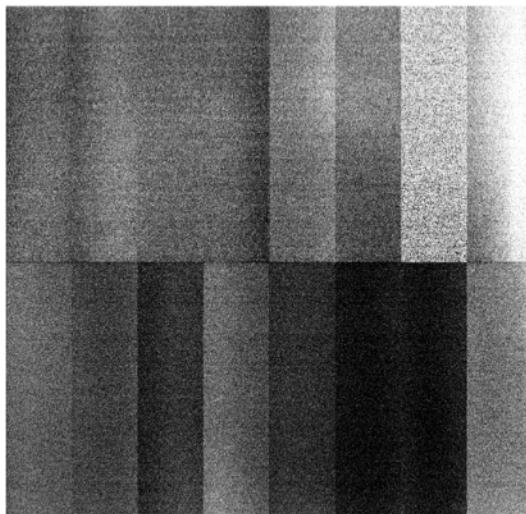
- postISRCCD images
 - all images are bias corrected and flat fielded
- Images shown represent the average of ten images taken over the span of ten minutes (with some exceptions)
 - Each individual image has 60 sec exposure time, except for 2:00 am images which have 120 sec exposure time
 - At 1:00 am only six out of ten images were able to be processed correctly
- All images were taken on one night (May 11) from about 8:00 pm to 4:00 am

Image processing procedure

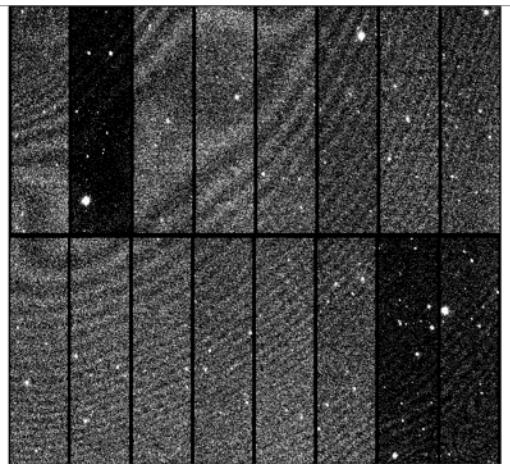
- Calculate master bias, master dark and master flat frame, use these three to process and produce resultant frame
- Master bias corrects for offsets, master dark corrects for dark current, and master flat helps correct for things like amplifier gain and unwanted static features

Master Dark

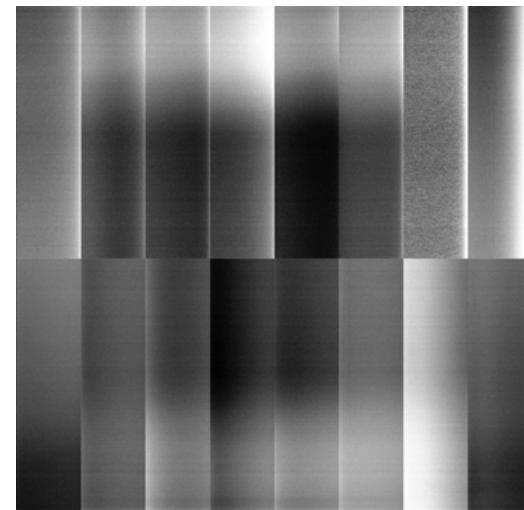
(not good, so it will be thrown out)



Raw image

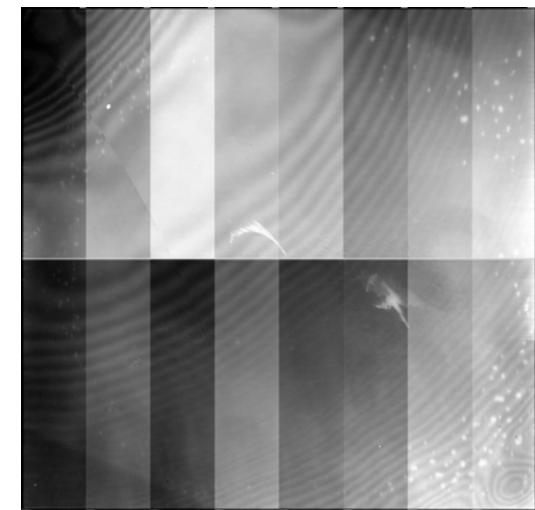


Master bias



Master Flat

(Fringing amplitudes <0.5% of median)

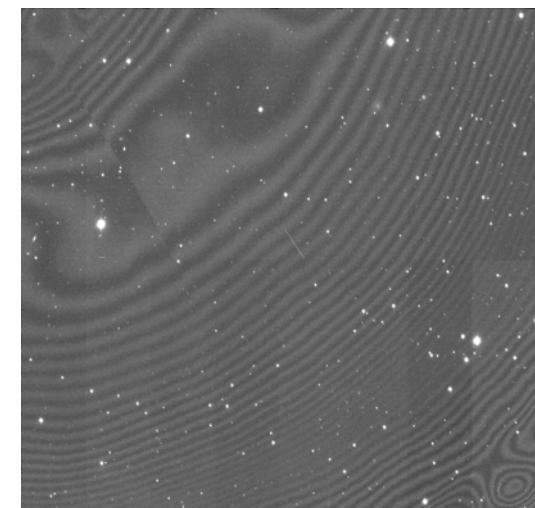


Processed Image

postISRCCD processing with Master bias and Master flat correction

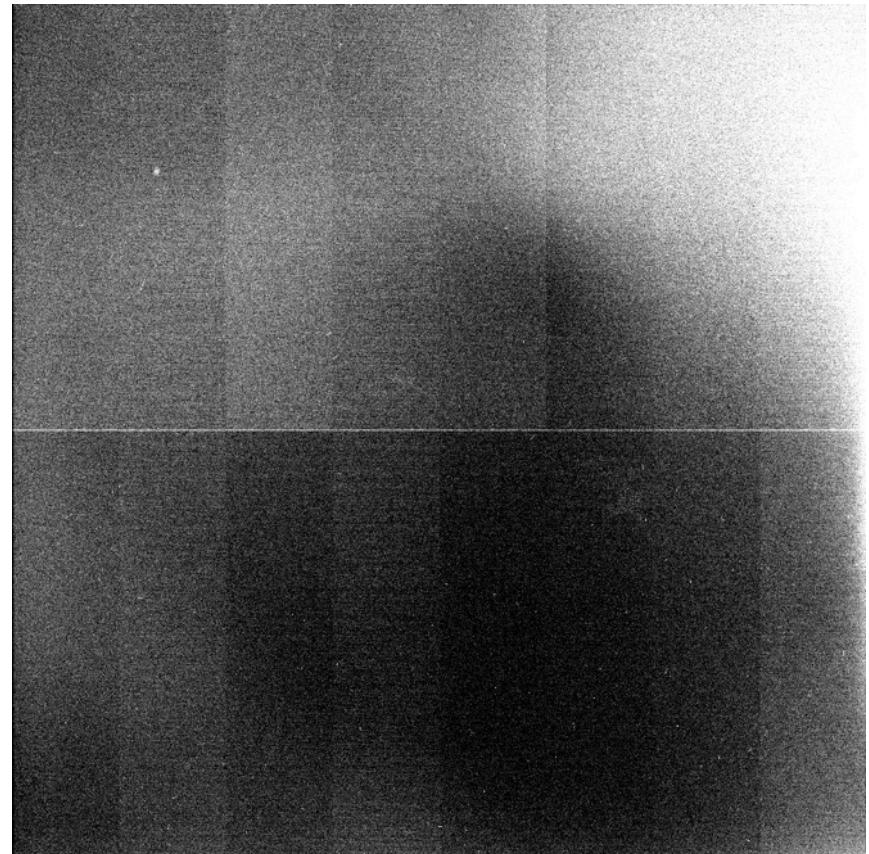


(Fringing amplitudes ~2-3% of median)



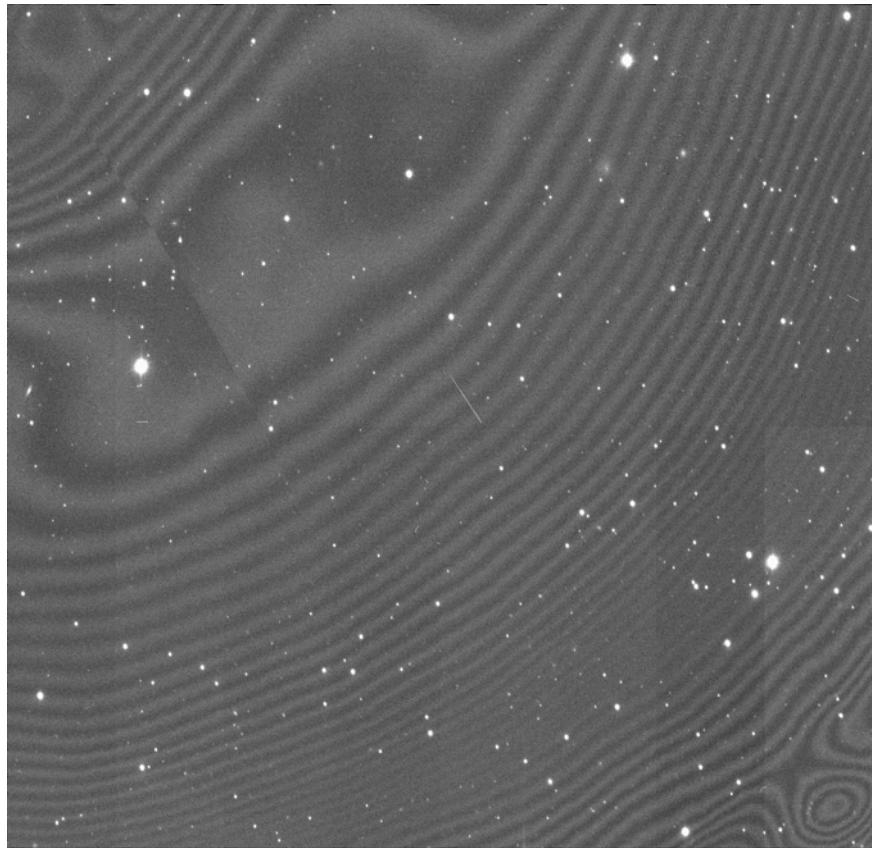
Processing issues

- Few bad darks ruined “dark subtracted” images
- Appears some light leaked into dark flats



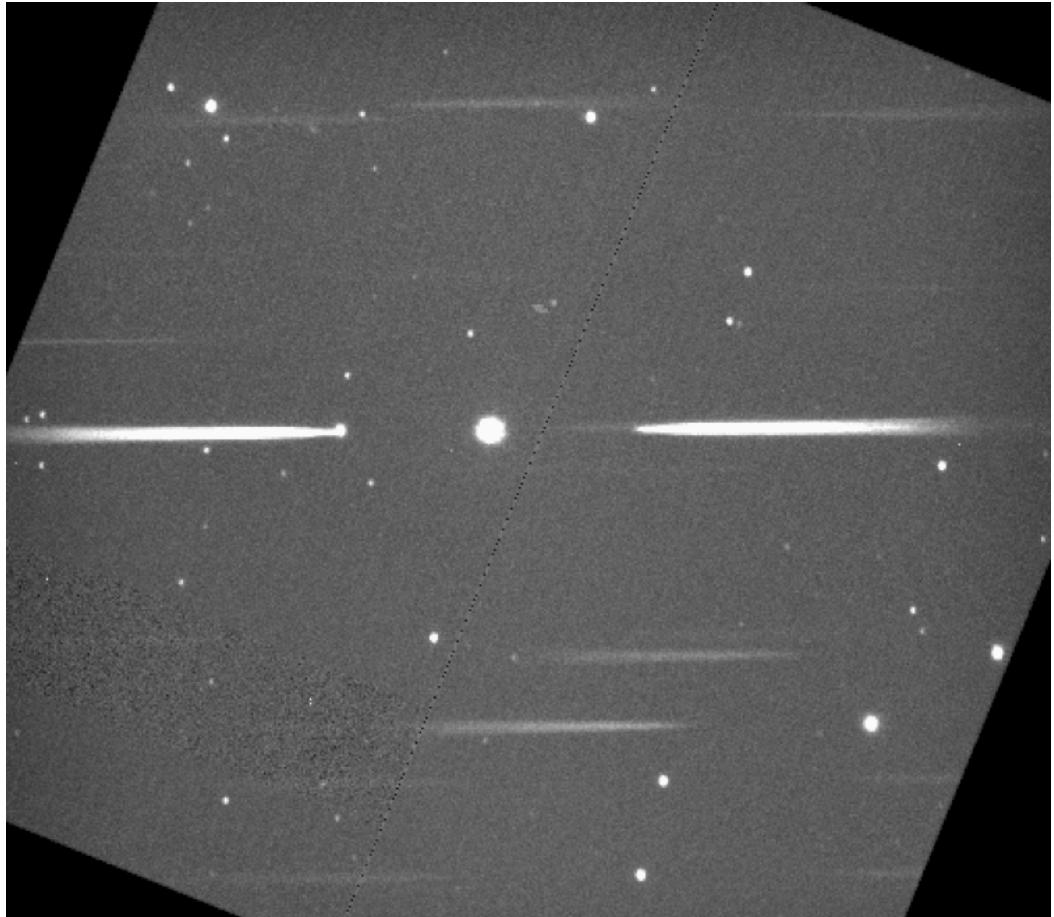
Processing issues contd.

- Processed image still has some identifiable amplifier regions
 - Bias corrected and flat fielded but no dark correction applied to image
- Could try to apply gains from Fe55 calibration, had good results in the past
- Does not affect the results below since D.Kirkby's code correct for this anyway



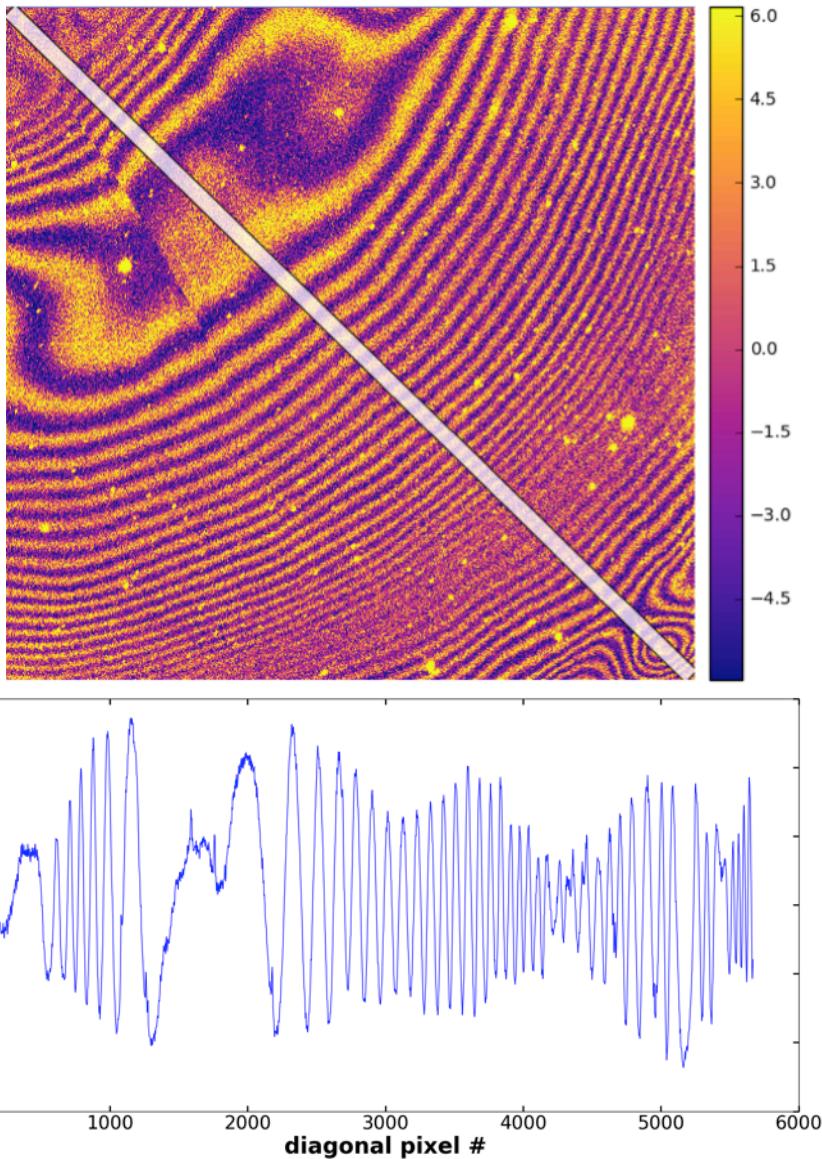
Fe55 gain corrected image

- A random Ronchi grating image
- Bias subtraction and Fe55 gains applied



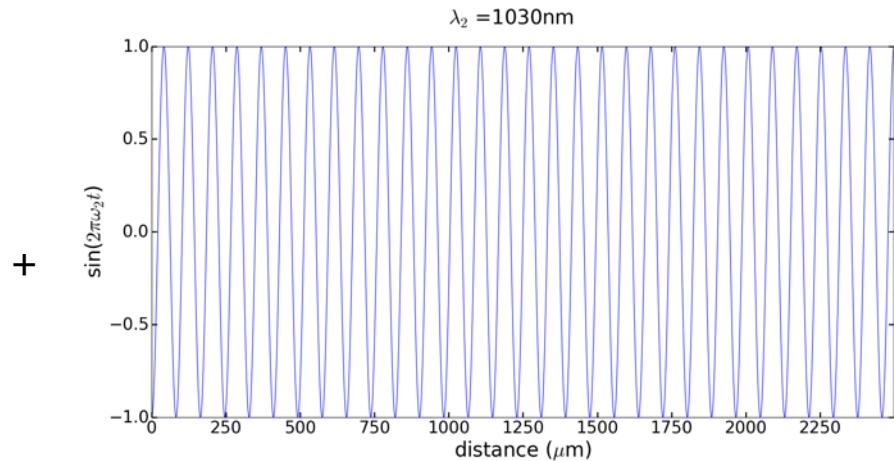
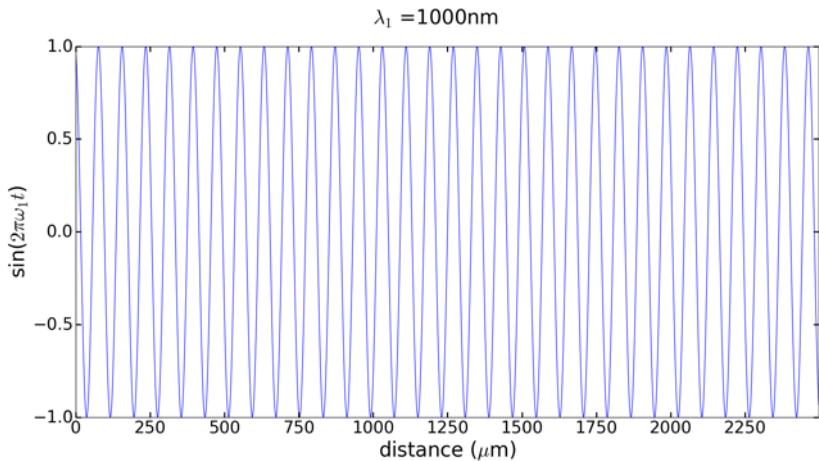
Woody Gilbertson

Slice plots of Monocam images

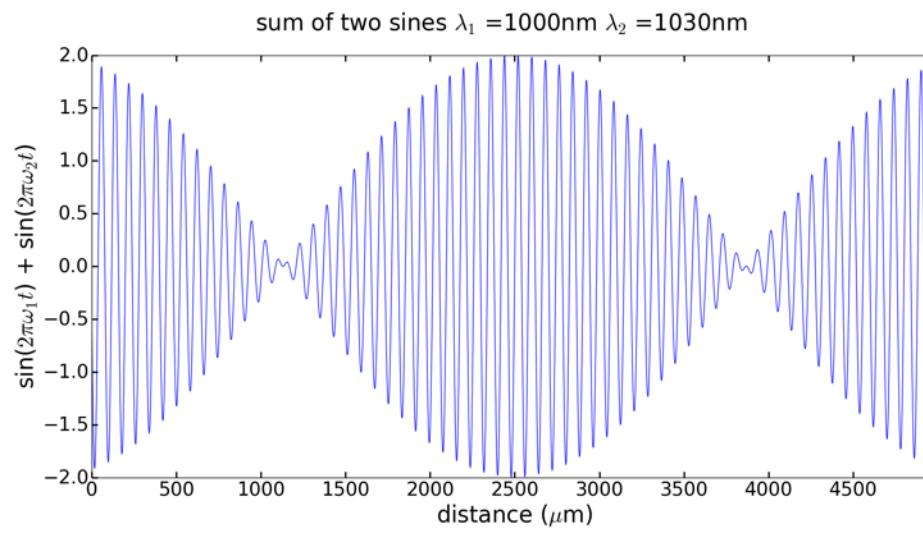


- Take slice plot extending from top left corner of each image to the bottom right corner
- Throw resulting slice plots into PCA code
- Slice plots described in more detail in [previous presentation](#) given on 11/3/2016

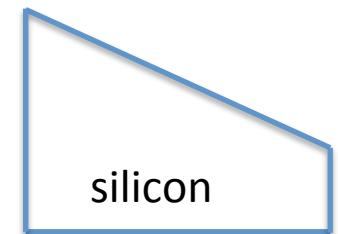
Sum of sines creates beats



=



Varying sensor thickness creates fringes

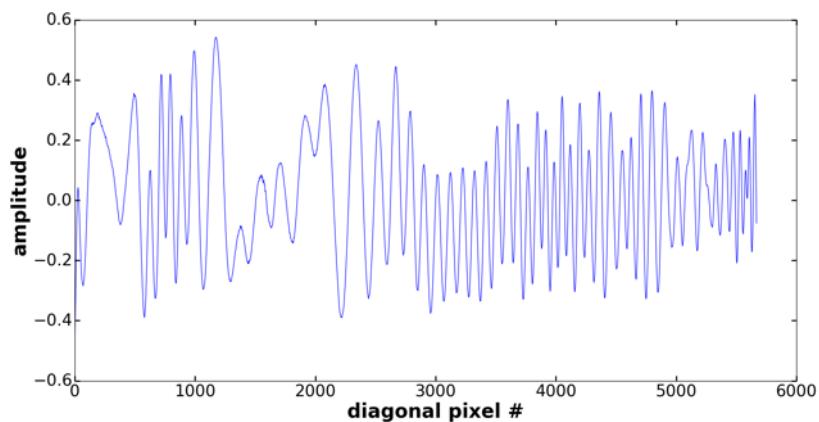


Lab measurements

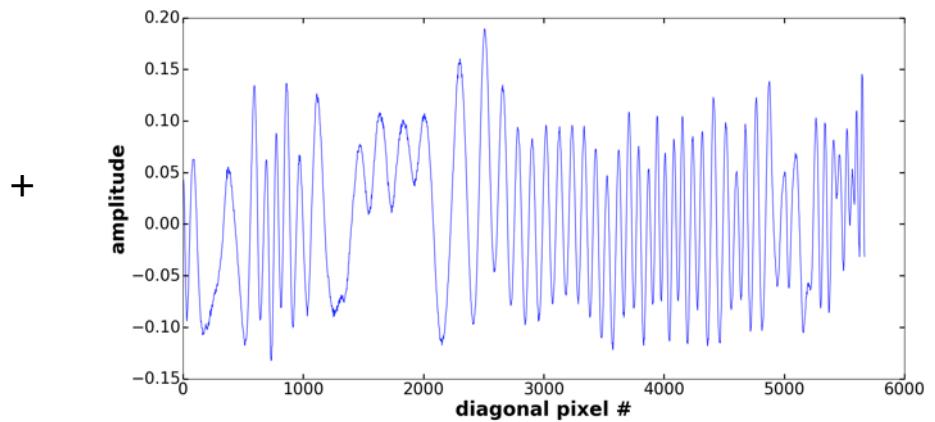
- We have several lab flats taken with the same sensor as Monocam
- Flats from 900nm to 1100nm in increments of 10nm will be used for the following analysis
- Each flat is first median subtracted using David Kirkby's code and then divided by its interquartile range
- Next, flats are multiplied by LSST Y4 throughput values

Sum of two lab flats creates similar node

1000nm flat

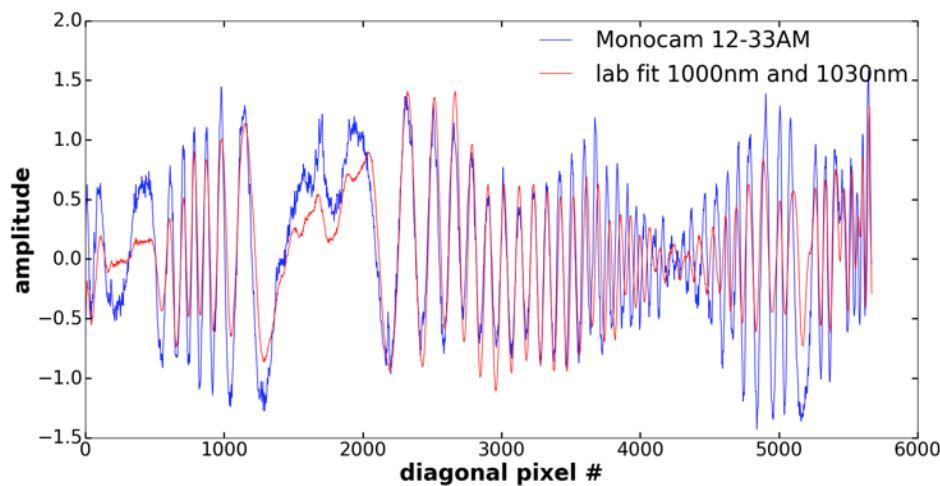


1030nm flat



+

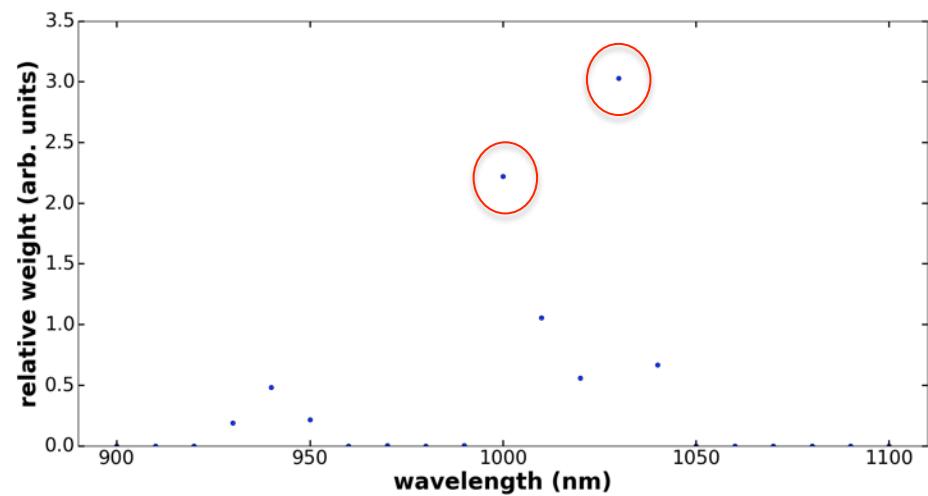
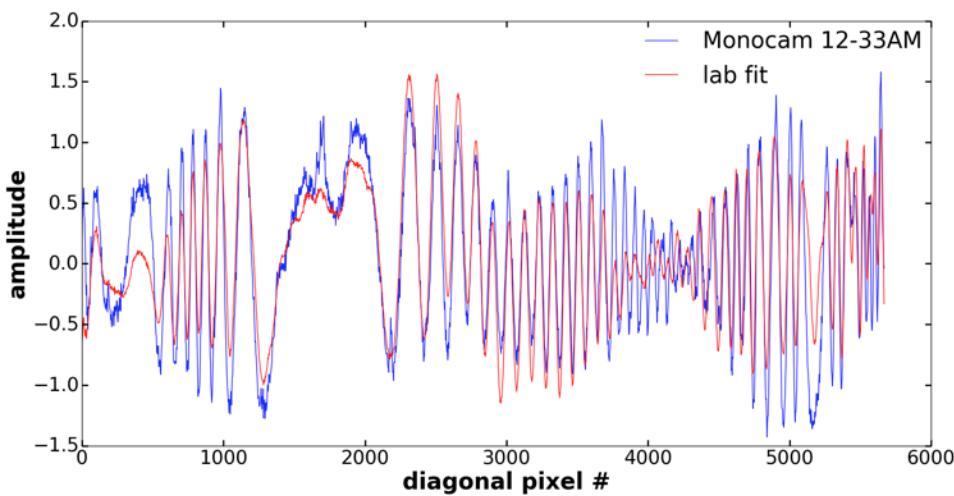
=



Lab comparison: sum of all flats with weights

Observed pattern = $\sum_{n=0}^{20} \alpha_n \Psi_n$

Where Ψ_n represents a lab flat taken in increments of 10 nm from 900nm to 1100 nm and α_n represents optimized parameter from fitting lab data to pattern



Principal Component Analysis: Procedure

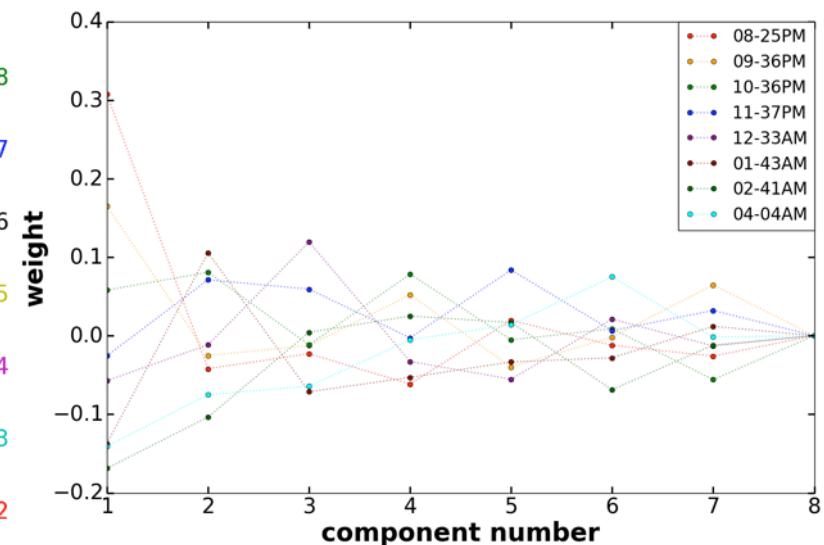
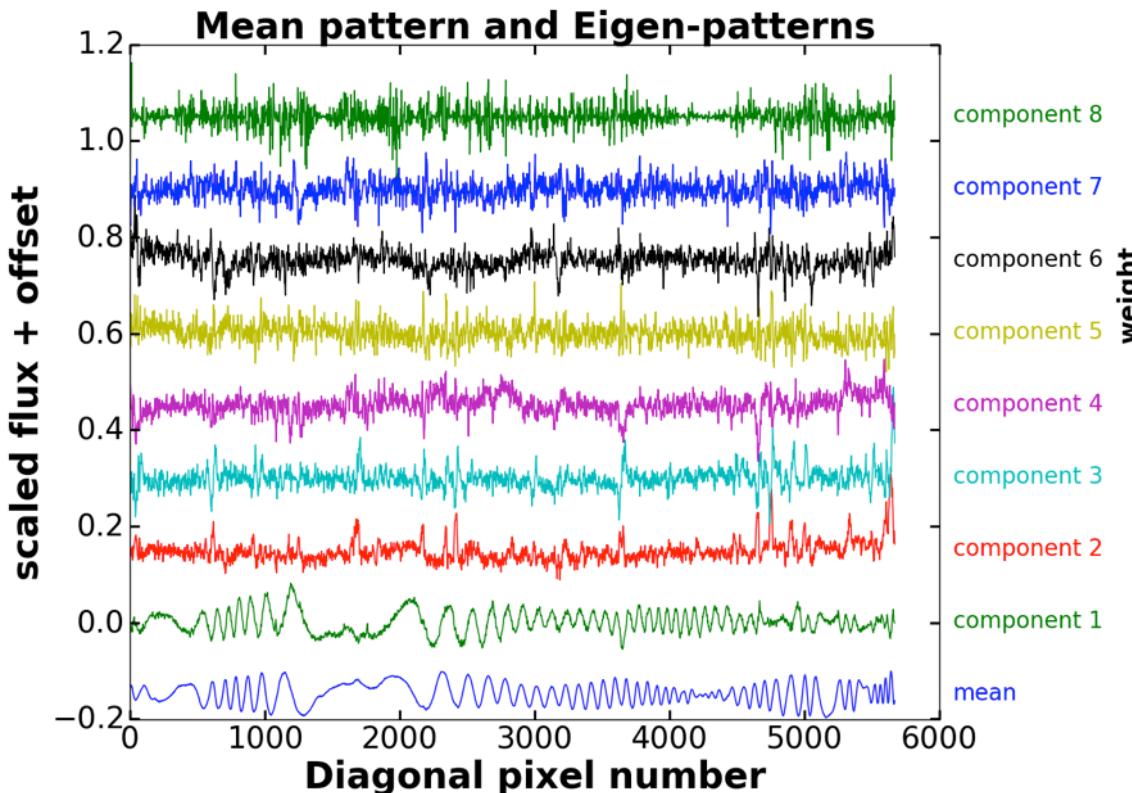
- Principal component analysis (PCA) tries to capture what elements in a set of data varies most

$$\text{fringe pattern} = \mu + \sum_{n=1}^8 w_n \chi_n$$

- Each hour of data can be represented as a sum of these principal components plus an average frame
- Find principal components using scikit-learn's code:
 - First principal component has highest variance, second principal component has second highest variance, and so on



PCA: Results

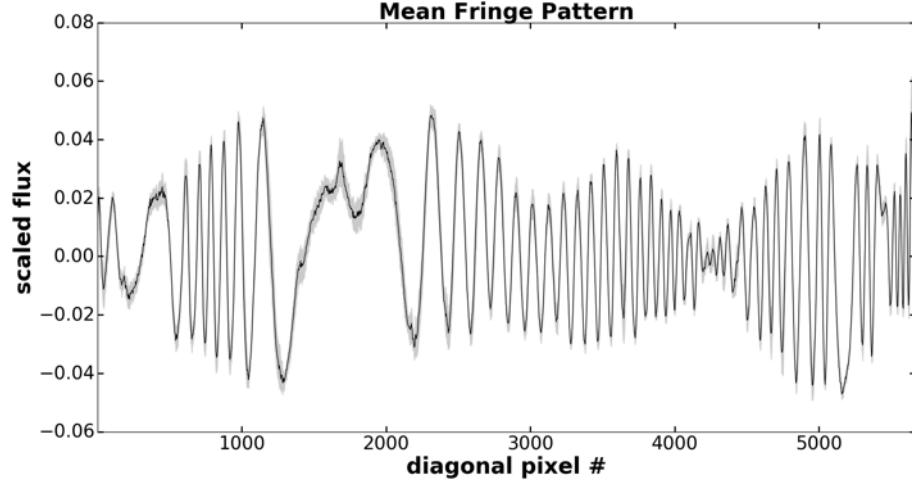


$$\text{fringe pattern} = \mu + \sum_{n=1}^8 w_n \chi_n$$

μ = mean pattern

χ_n = n^{th} principal component (Eigen-pattern)

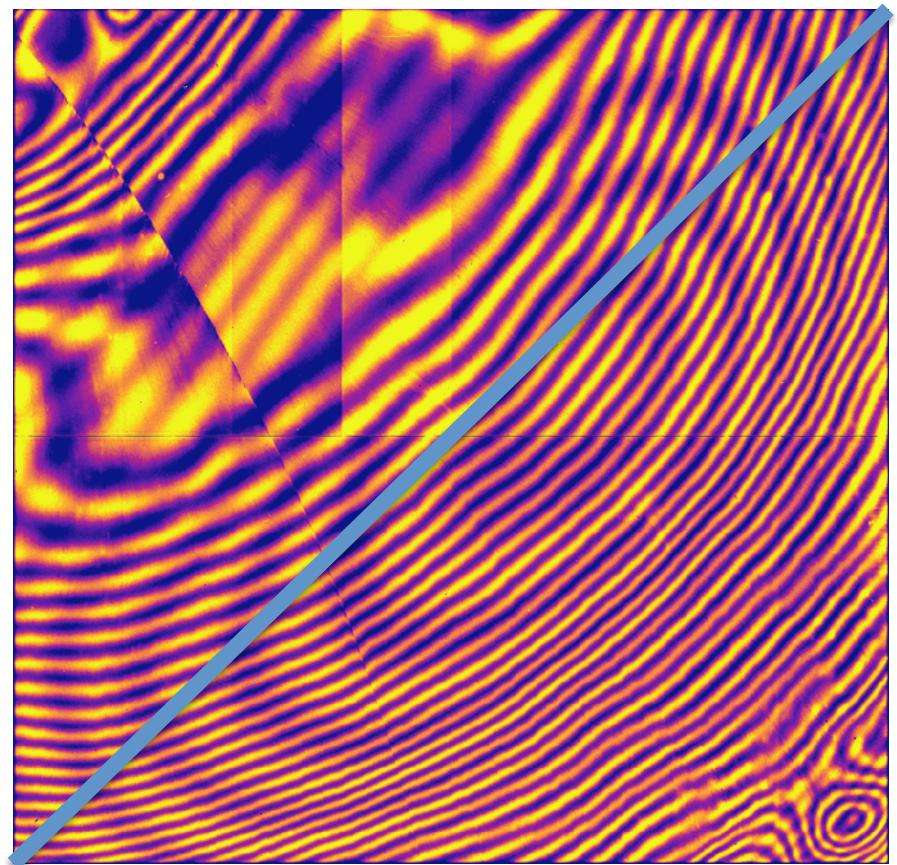
w_n = n^{th} weight factor



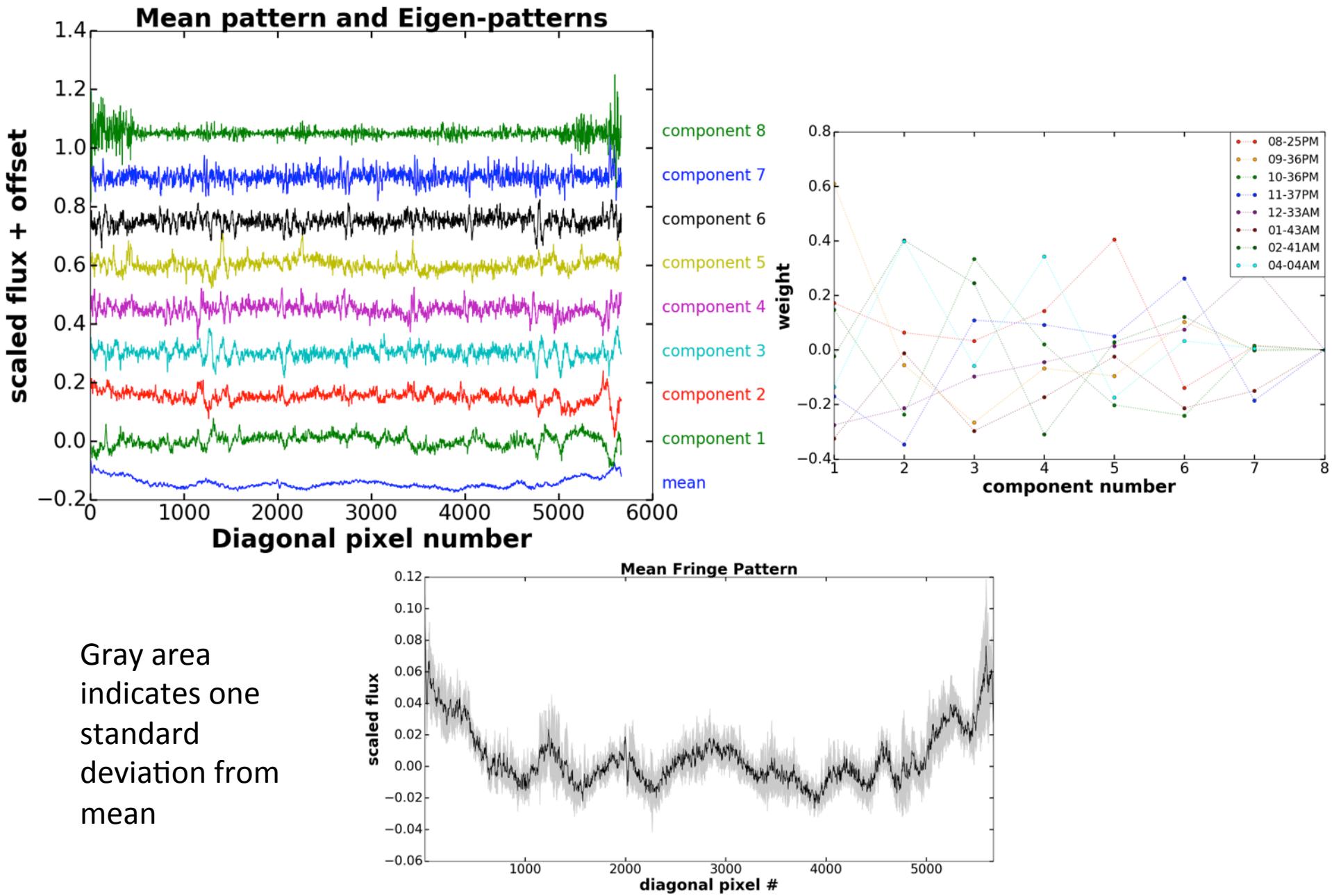
Gray area indicates one standard deviation from mean

Secondary fringe pattern?

- Secondary fringe pattern easily visible in lab data, though harder to see in Monocam data
- 1020 nm lab flat with visible secondary pattern shown to right for reference



Slice plot across opposite diagonal

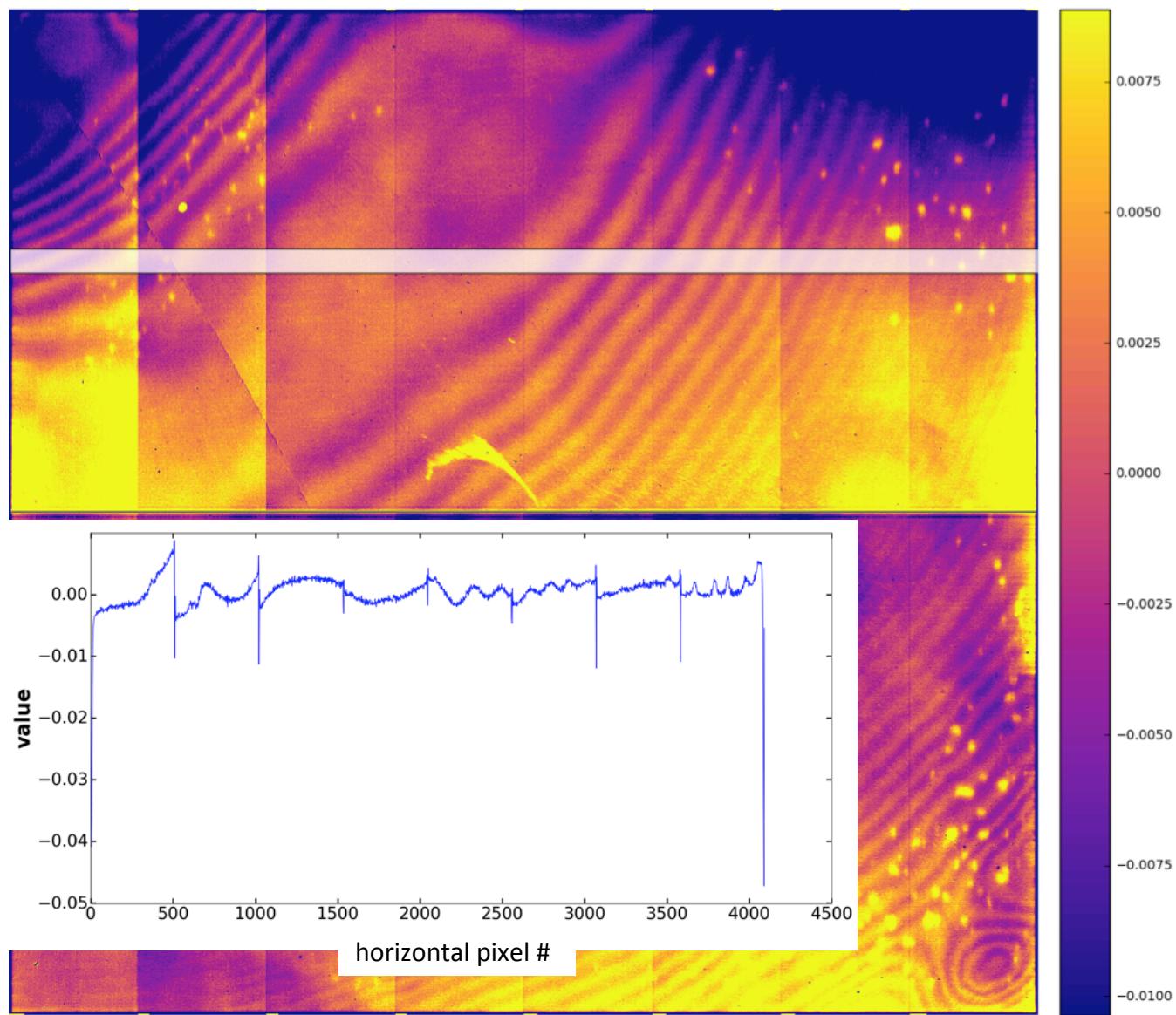


Summary

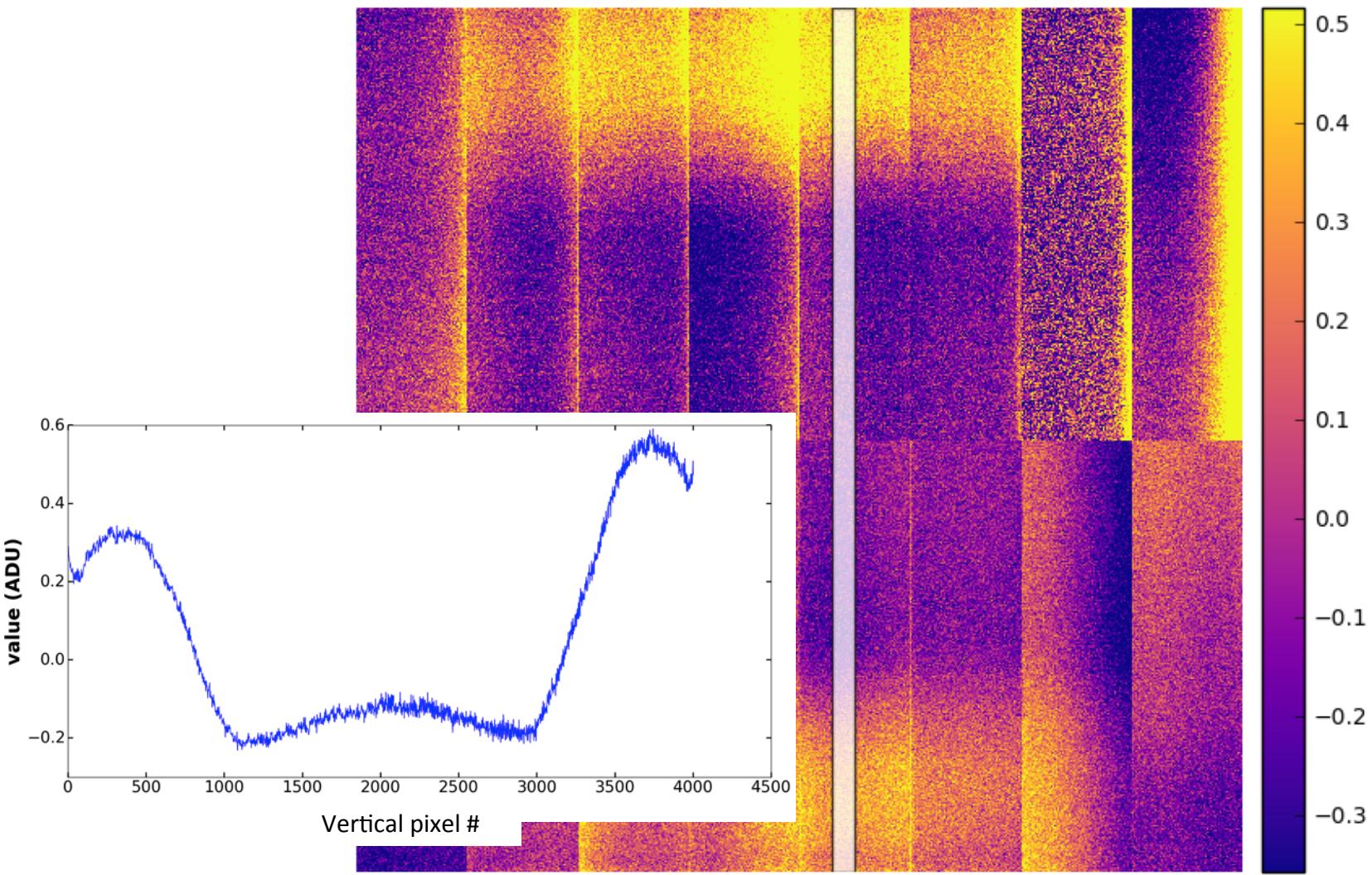
- Re-processing images using 12.1 version of stack still leaves some amplifier boundaries visible
- Lab fit to midnight pattern indicates 1000 and 1030 nm OH lines play large role at that hour
 - Fits with lab data are very rough, and results from fit should be taken with a grain of salt
 - Secondary fringe seen?
- Currently being written up in PACCD proceedings

Backups

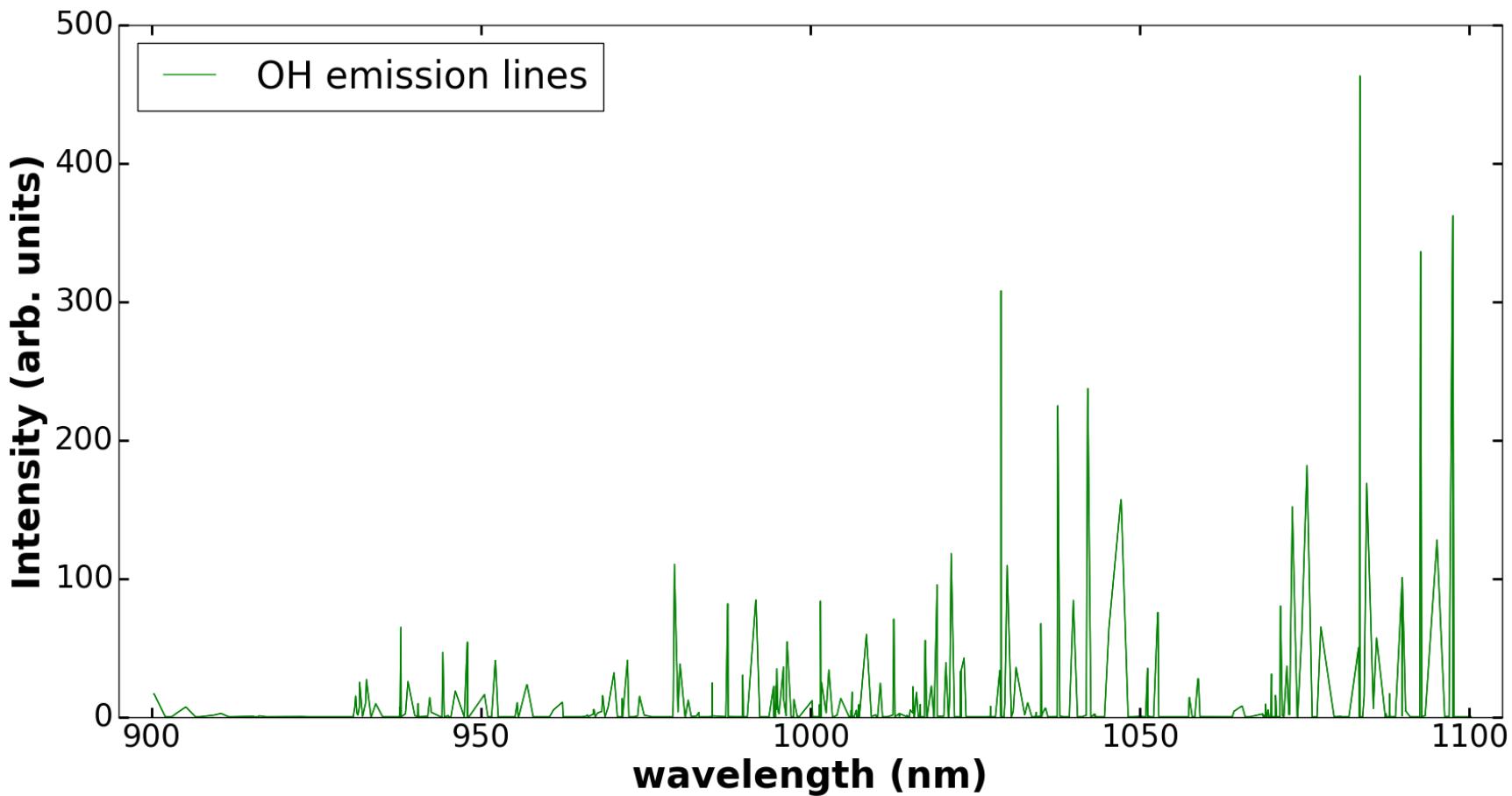
Master flat



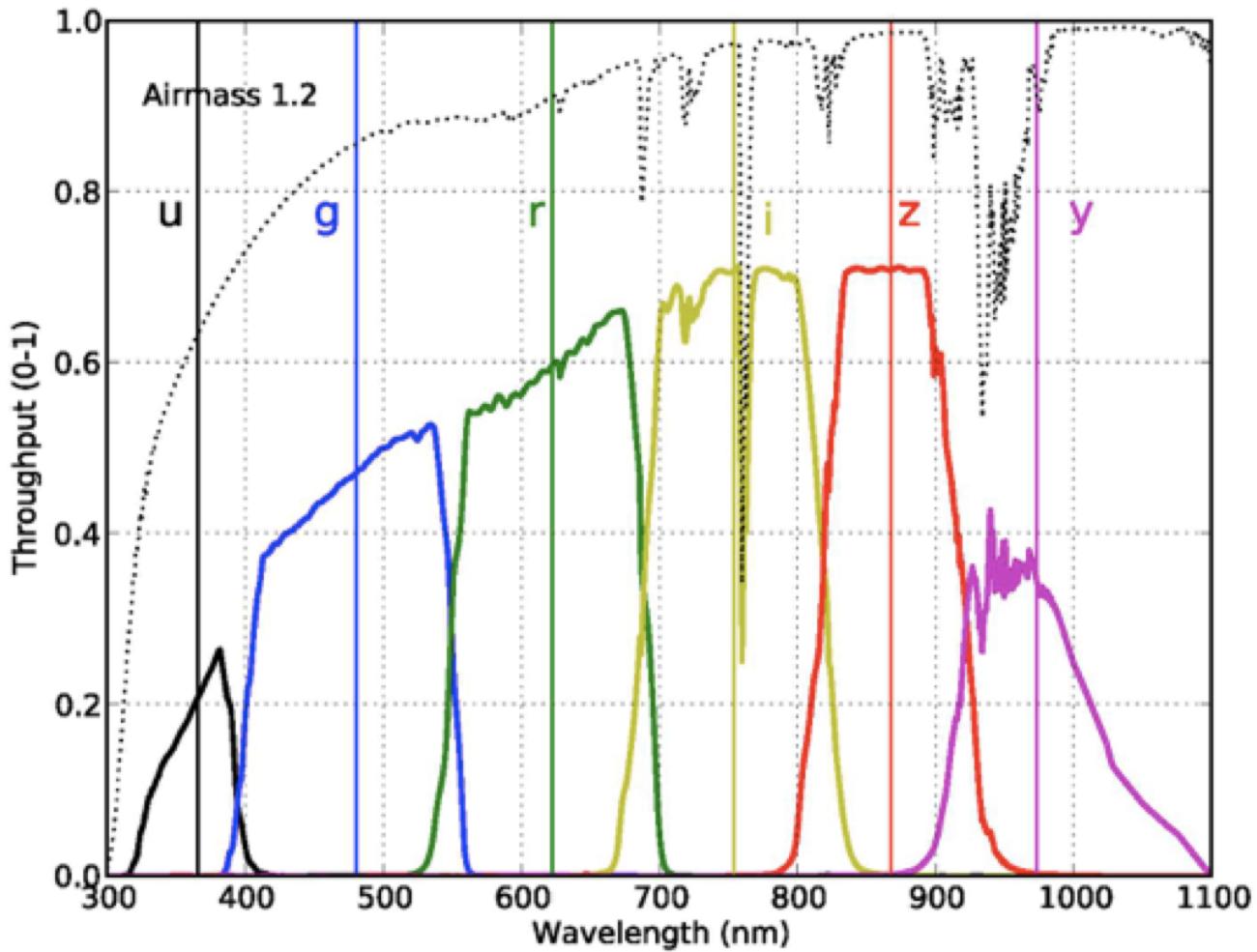
Master bias



OH emission spectral lines



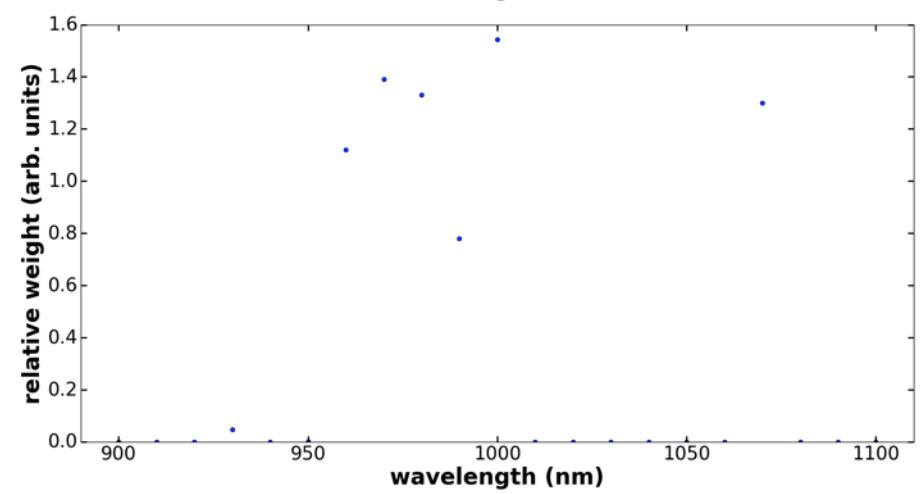
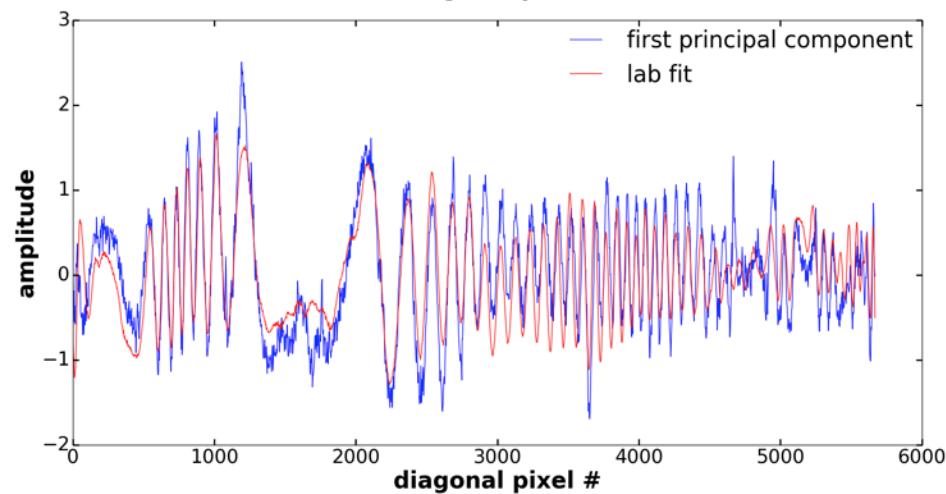
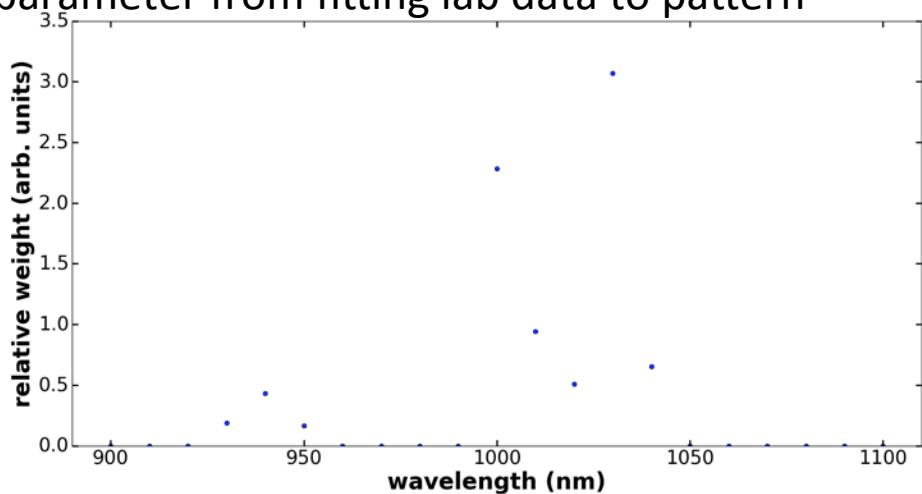
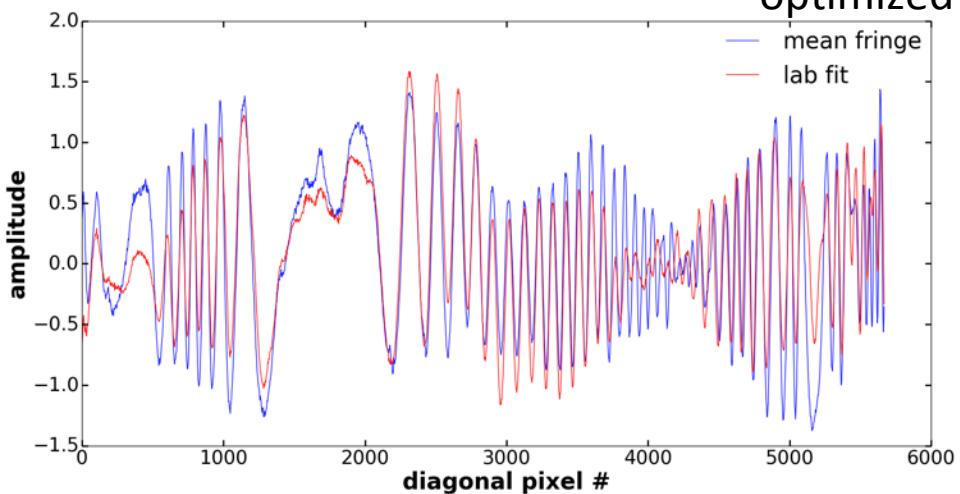
LSST Filter Throughput



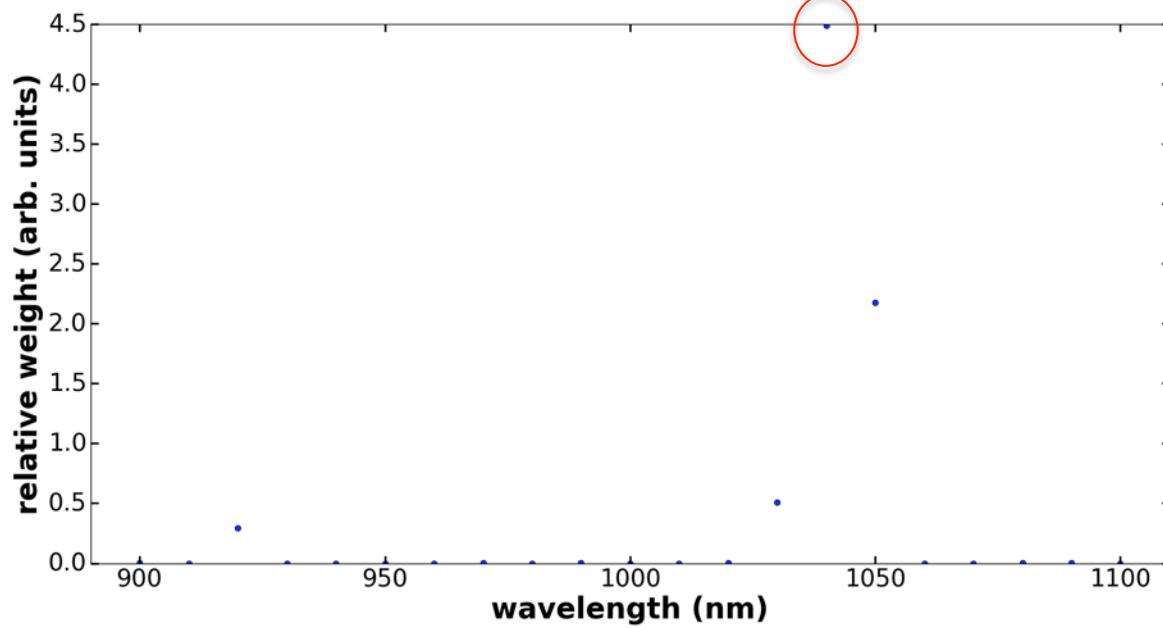
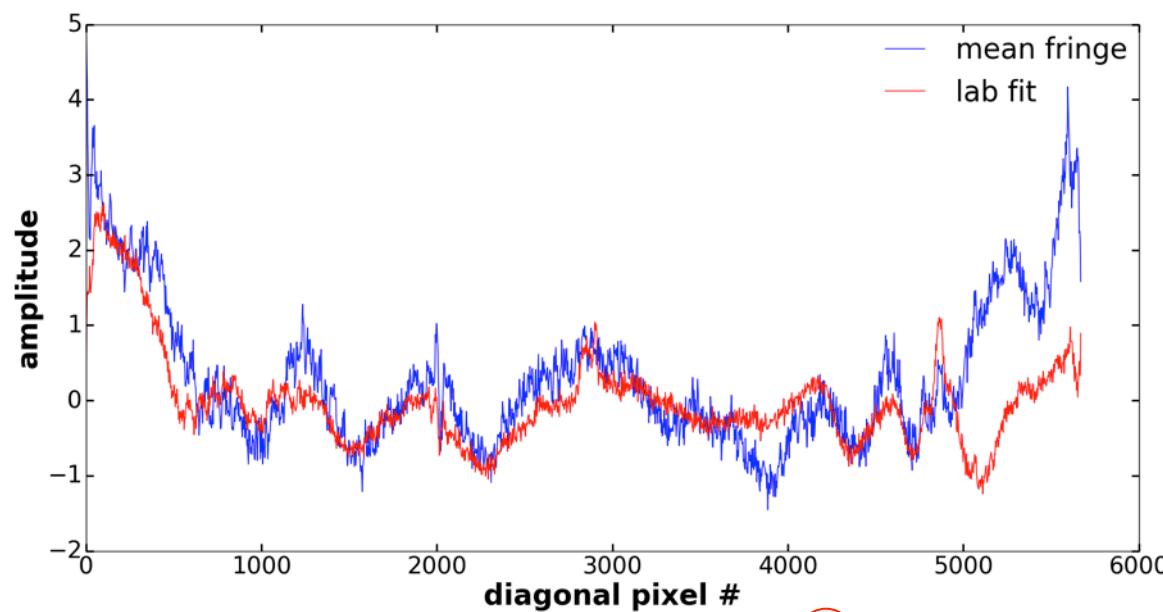
PCA: Lab comparison

Mean or PCA pattern = $\sum_{n=0}^{20} \alpha_n \Psi_n$

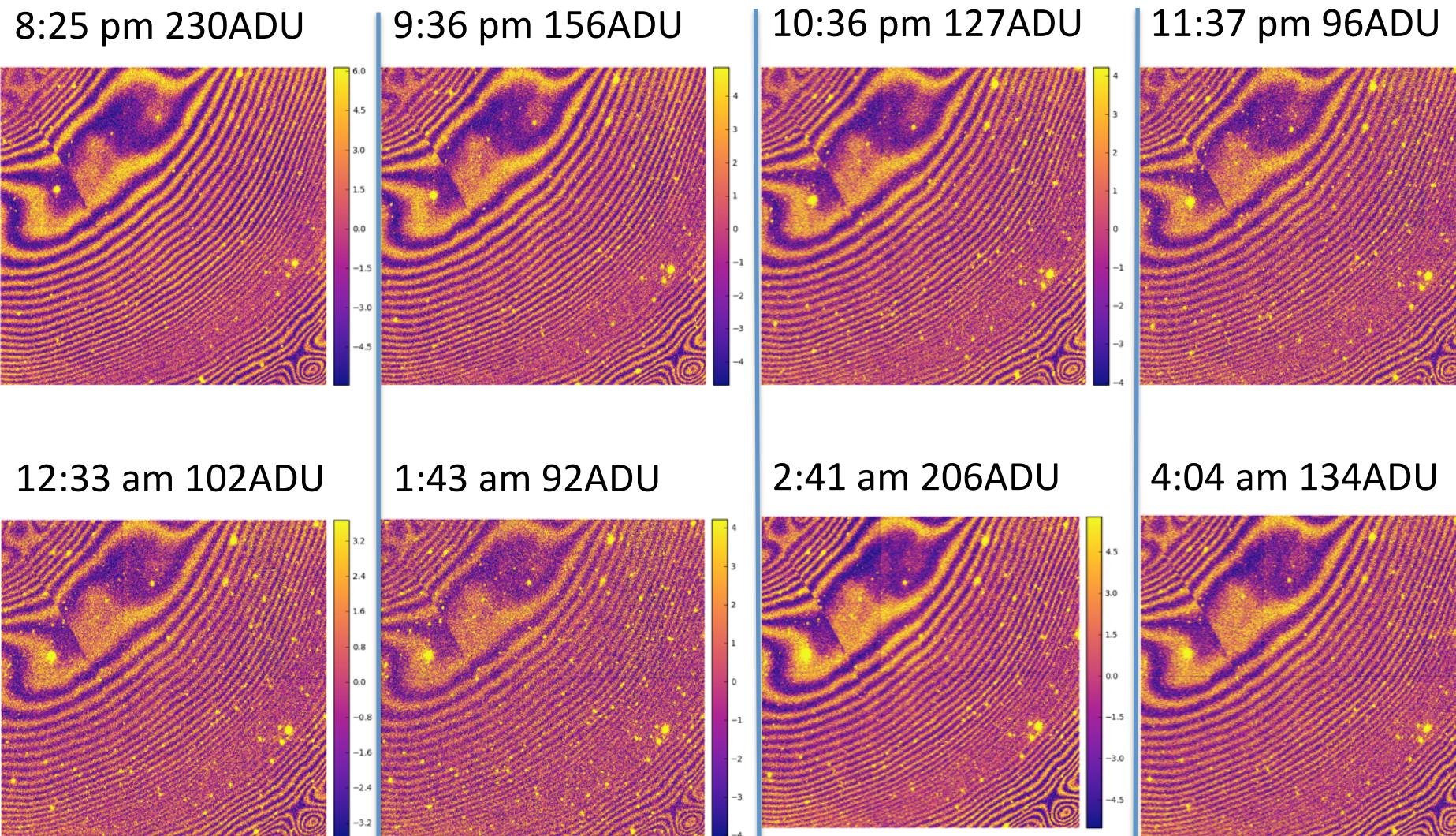
Where Ψ_n represents a lab flat taken in increments of 10 nm from 900nm to 1100 nm and α_n represents optimized parameter from fitting lab data to pattern



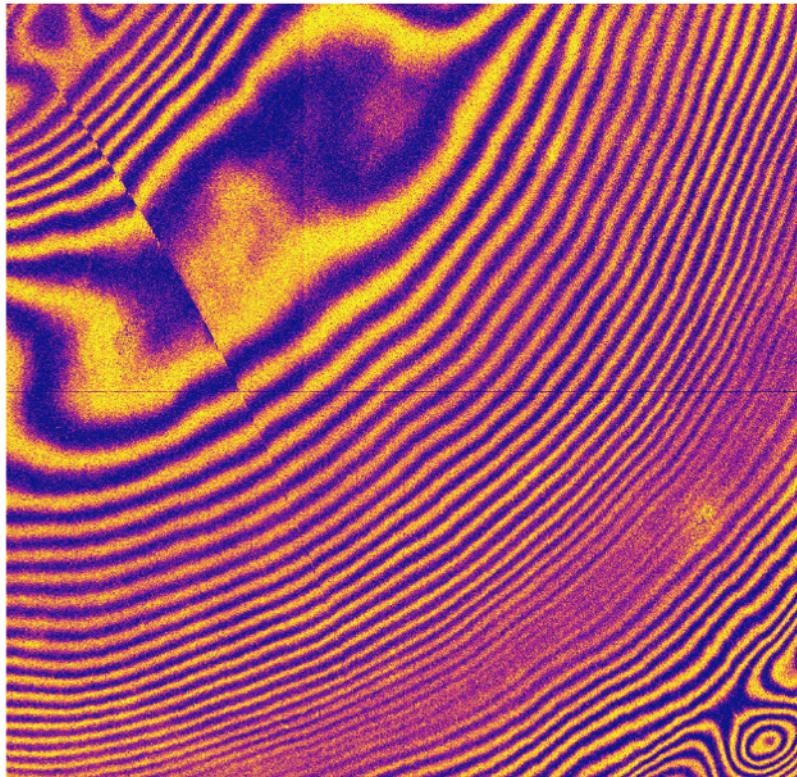
Opposite diagonal PCA: lab comparison



All re-processed averaged raw images with median signal



Star subtraction

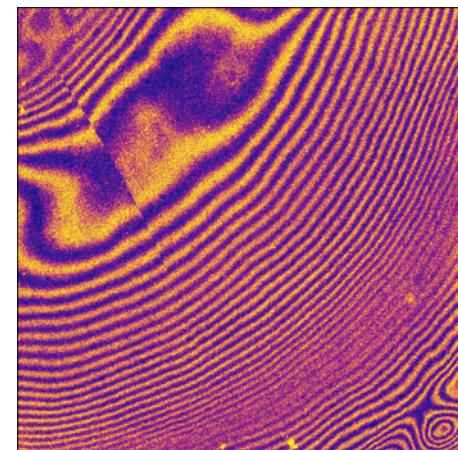


- Mean fringe pattern (shown on left) is subtracted from hourly raw images
- We add an adjustable constant β to be multiplied by mean fringe in order to optimally reduce fringe amplitudes
- At 11 o'clock fringes vanish almost entirely
- Trace contours around stars in this image and shift contours around to match stars in other image
- Fill in contours with min value (times a correction factor to keep values consistent withh image) from all other frames in that region

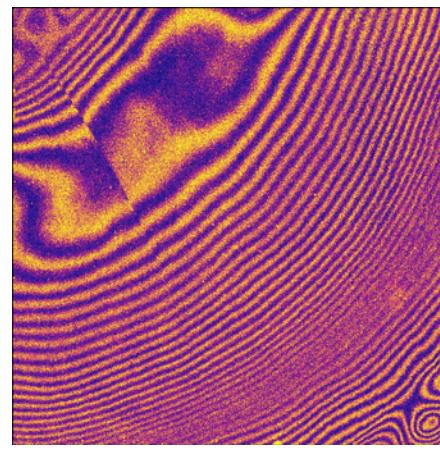
Fringe corrected pattern = raw average pattern – β^* (mean fringe)

All averaged raw images star subtraction

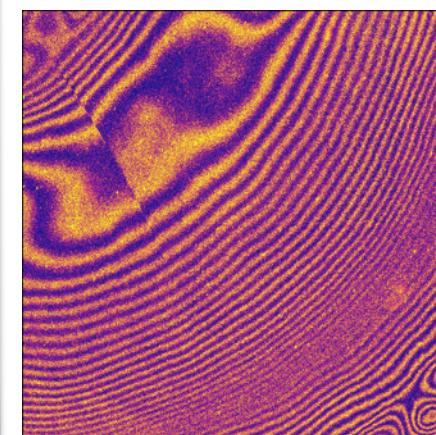
8:25 pm 230ADU



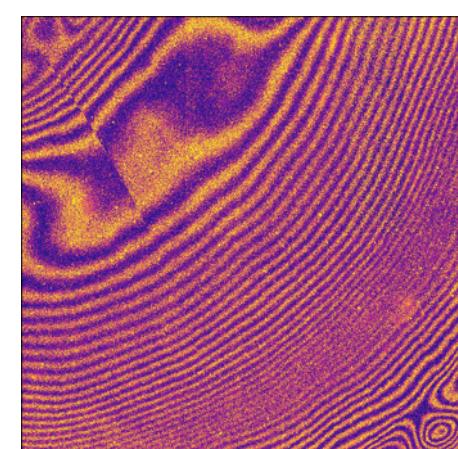
9:36 pm 156ADU



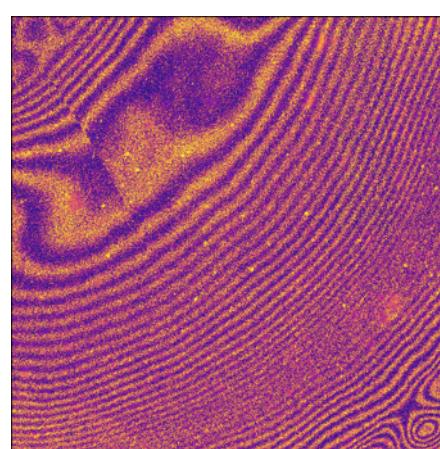
10:36 pm 127ADU



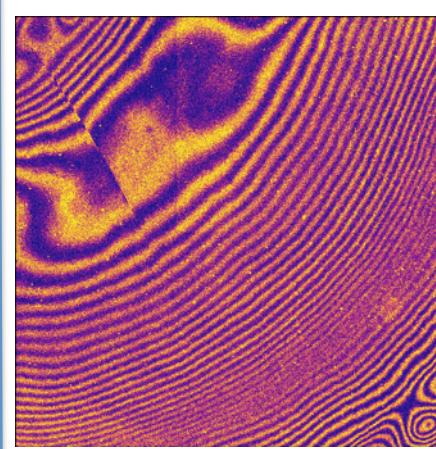
12:33 am 102ADU



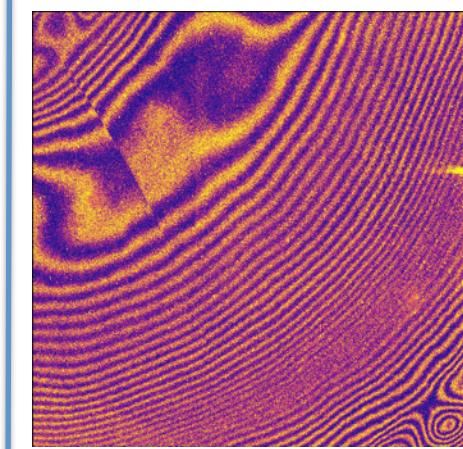
1:43 am 92ADU



2:41 am 206ADU

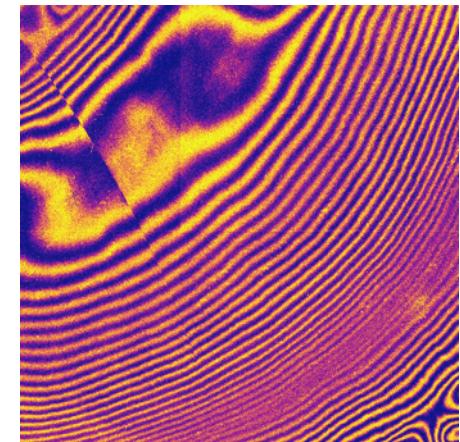


4:04 am 134ADU

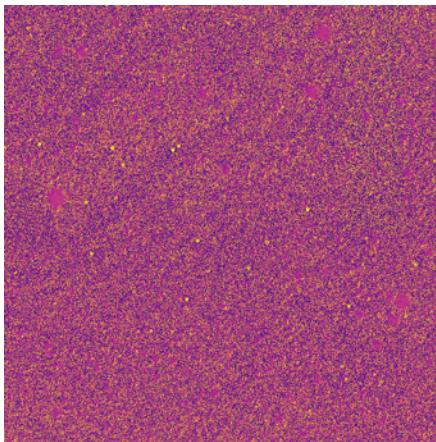


2-D PCA

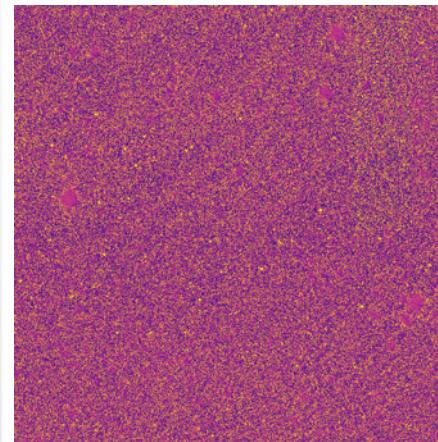
mean



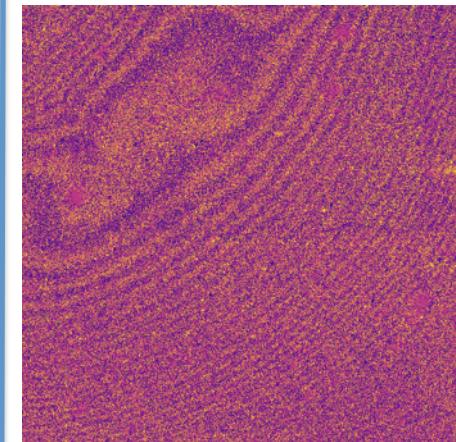
Component 1



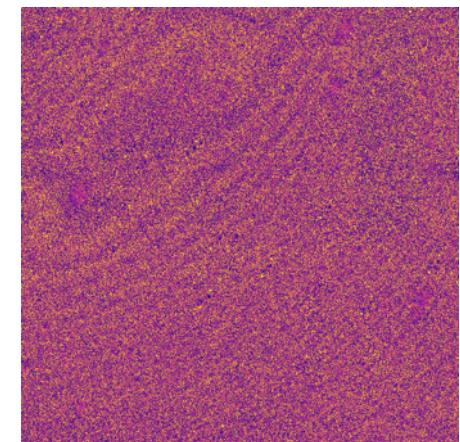
Component 2



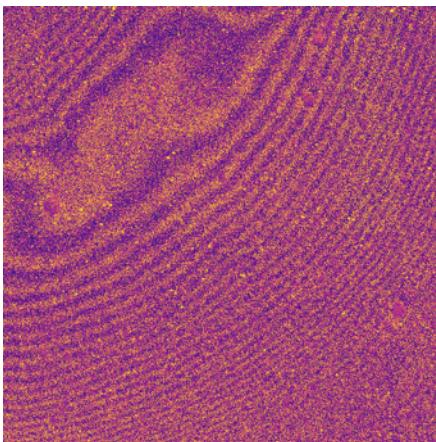
Component 3



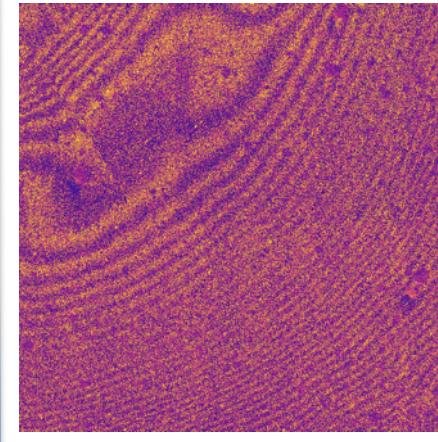
Component 4



Component 5



Component 6



Component 7

